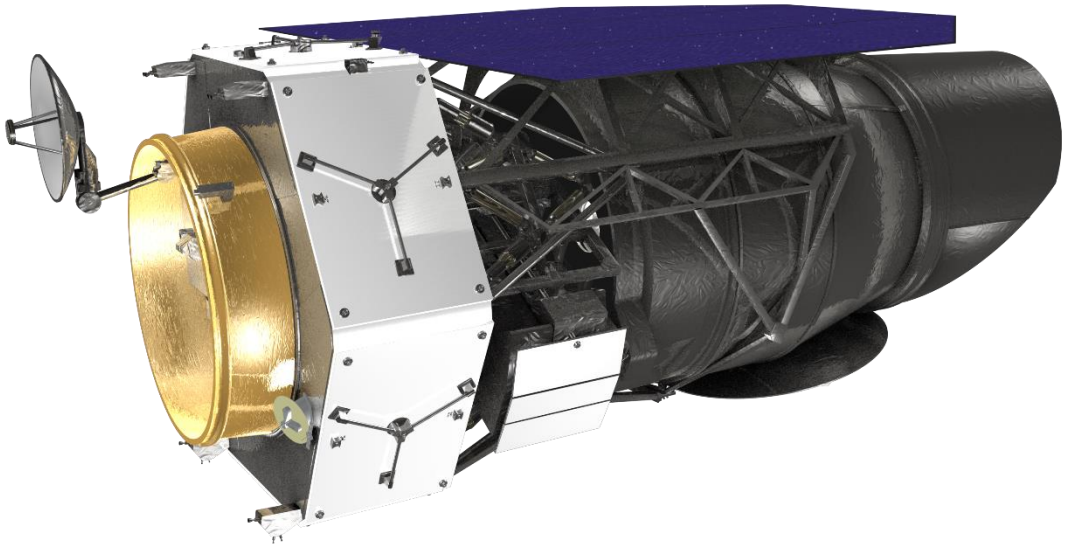
Jet Propulsion Laboratory
California Institute of Technology

IMPipeline: An Integrated STOP modeling pipeline for the WFIRST coronagraph

ASME Verification and Validation Symposium
May 4 2017



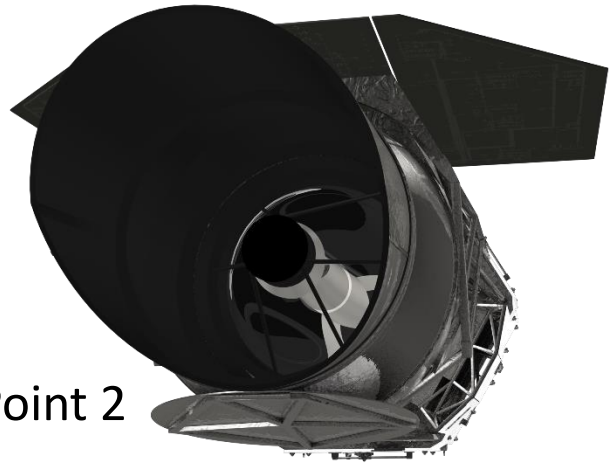
Navtej Saini, Kevin Anderson, Zensheu Chang, Gary Gutt, Bijan Nemati
NASA JPL, California Institute of Technology, Pasadena, California

Talk Outline

1. Introduction and Motivation
2. Fundamentals of STOP Analysis
3. WFIRST Integrated Modeling Pipeline
4. Verification and Validation of STOP Model
5. Preliminary Results
6. Future Work

Introduction

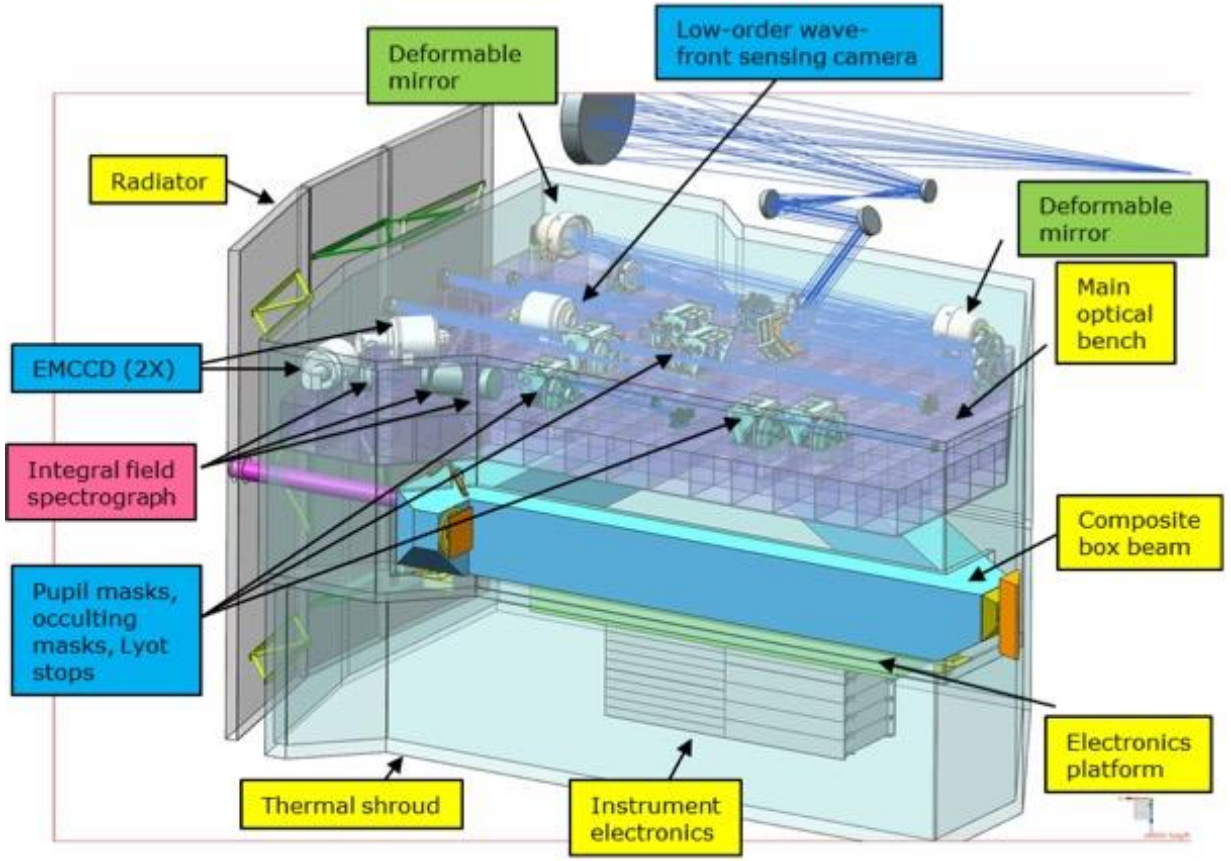
WFIRST, the **W**ide **F**ield **I**nfra**R**ed Survey **T**elescope, is a future NASA telescope to explore exoplanets and answer essential questions in the field of dark Energy and infrared astrophysics.



- 2.4 meter diameter primary mirror
- Designed for 6 year mission
- Will be located at the Earth-Sun Lagrange Point 2
- Two instruments onboard – Wide Field imager (WFI) and a Coronagraph (CGI)
- WFI will provide capabilities to perform the Dark Energy, Exoplanet microlensing, and NIR surveys
- CGI supports the Exoplanet high contrast imaging and spectroscopy science

Credit: NASA GSFC (https://wfirst.gsfc.nasa.gov/gallery-full.html)

WFIRST Coronagraph



- Coronagraph imager covers spectral range of 0.43 – 0.98 µm,
- Two deformable mirrors form a sequential wavefront control system (WFCS) that compensates for both phase and amplitude errors in the telescope and coronagraph optics
- The spectroscopy mode uses and integral field spectrograph (IFS)

Image credit: NASA GSFC (https://wfirst.gsfc.nasa.gov/observatory.html)

Fundamentals of STOP Analysis

- Thermal deformation is one of the major sources that can degrade the performance of a space telescope or antenna.

- Structural Thermal Optical Performance (STOP) analysis is the tool to assess the performance of a space instrument under thermal loads.

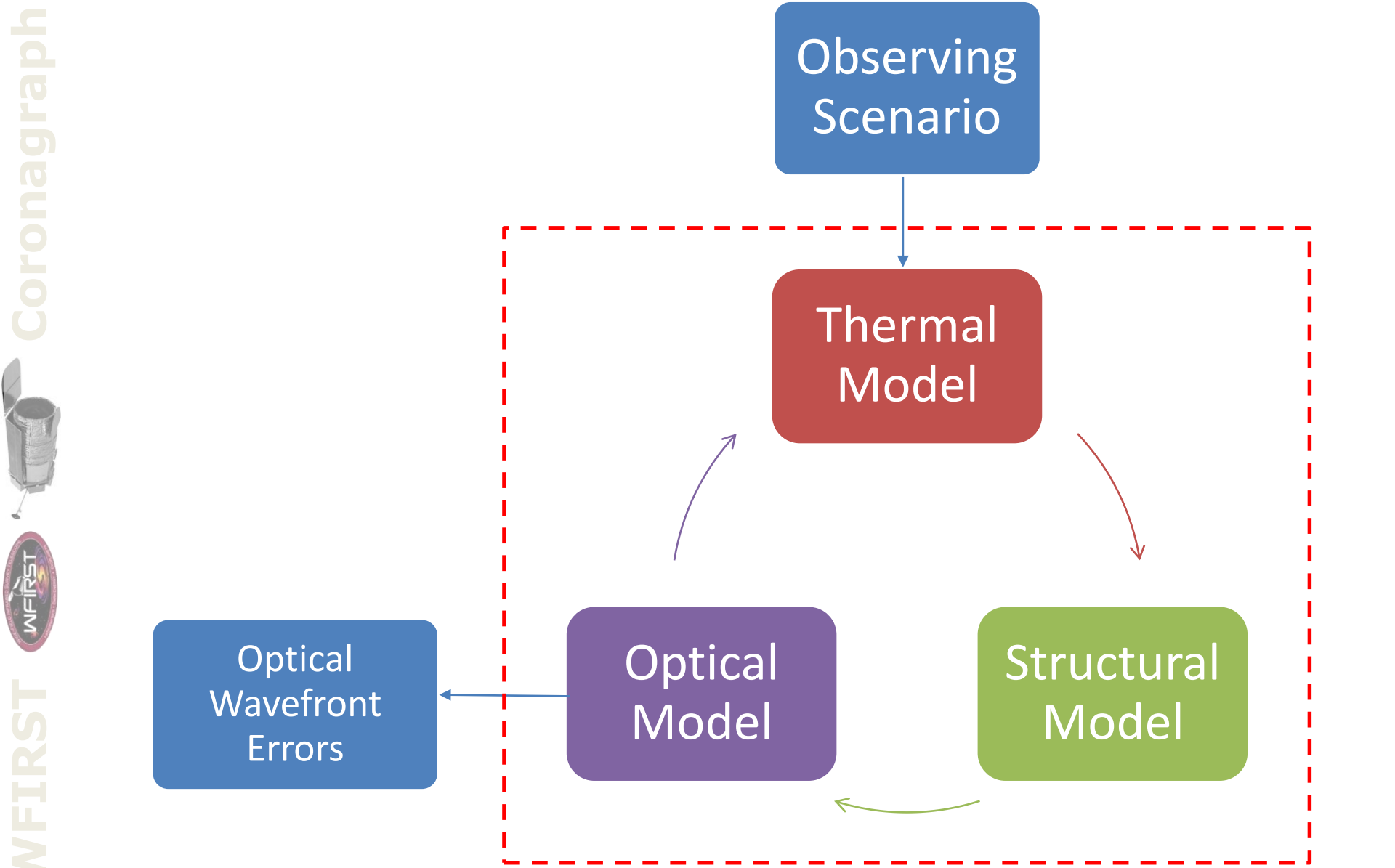
- STOP analysis includes
  - ✓ Thermal analysis
  - ✓ Structural (thermo-elastic) analysis
  - ✓ Optical analysis

- Structural model for STOP analysis
  - ✓ Output request: displacements only. Requires relatively coarse mesh, compared to models used for stress analysis.
  - ✓ A lot of analysis to perform.
  - ✓ Try to minimize the size of the STOP model
  - ✓ Can be used for other analysis, i.e., modal analysis, jitter analysis, random vibe analysis

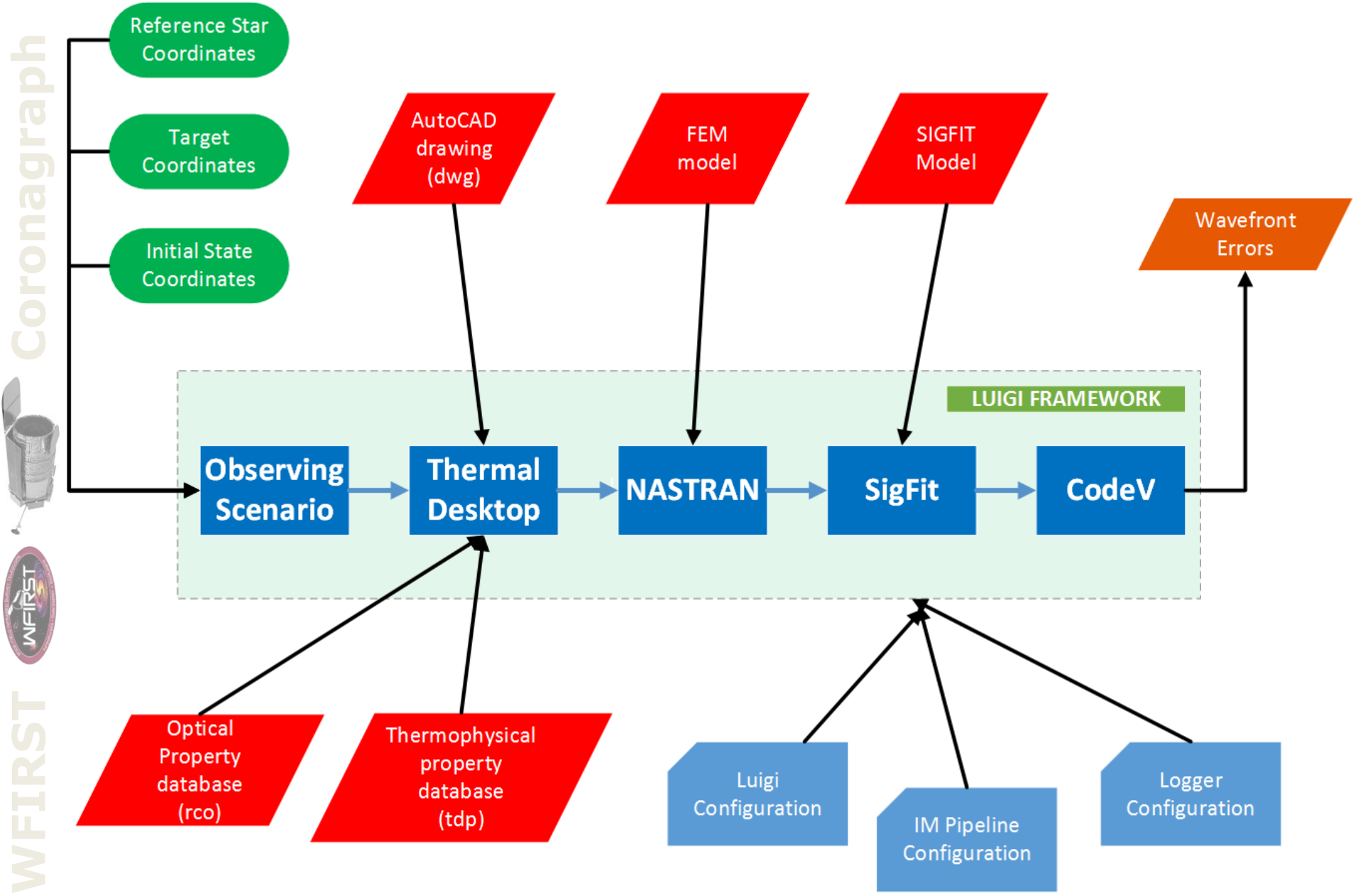
- Due to the requirement for high accuracy, temperature dependent CTE is used for calculating thermal loads

- Thermal loads are driven by environmental (orbital mechanics) conditions, the thermal control system must ensure that the componentry such as the Coronagraph Instrument (CGI) hold requirements on set-point with ample stability

Motivation


Observing Scenario
Thermal Model
Optical Model
Structural Model
Optical Wavefront Errors

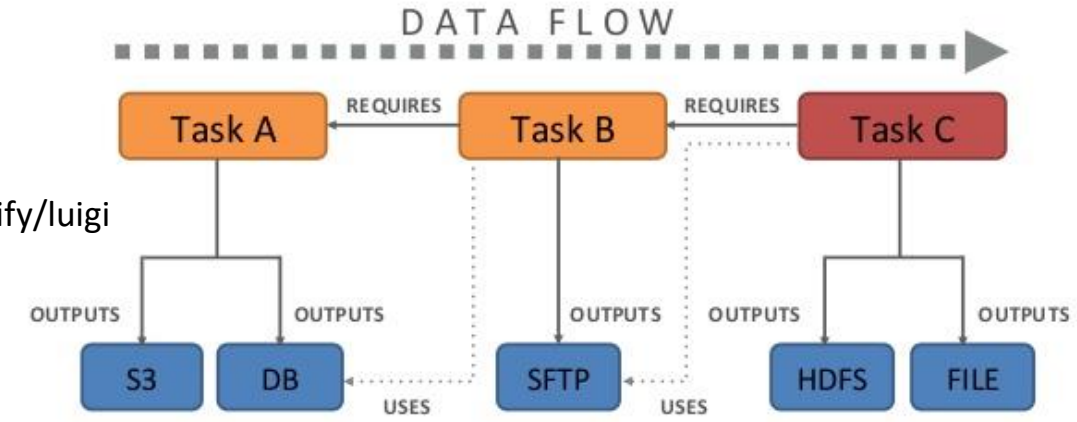
IMPipeline Block Diagram



Luigi Pipeline Framework

World's second most famous plumber

Source code: https://github.com/spotify/luigi



- Python framework to build complex pipelines of batch jobs
- Developed by engineers at Spotify and made open source for wider community
- Luigi server comes with a web interface to visualize, search and filter tasks
- Luigi is customizable, and handles dependency resolution, workflow management, visualization etc.

Observing Scenario

- Solar heating is a major input to WFIRST coronagraph thermal model
- The thermal model treats Solar radiation as plane wave incident on the spacecraft. We have to determine direction of the plane wave and the intensity of the flux
- During operations, the spacecraft attitude will be a function of the stars WFIRST will be observing, and the time of year during which they are observed.
- To simulate thermal effects of an observing scenario, we need to specify relation between the WFIRST spacecraft and the Sun.
- As WFIRST is expected to be at the Sun-Earth Lagrange point, Earth's infrared radiation is neglected.

Thermal Desktop

- Thermal Desktop is a PC CAD-based thermal model builder.
- Only runs on Windows OS and is integrated with AutoCAD.
- It creates the node and conduction network, launches SINDA/FLUINT for the solution, and provides post processing results.
- It interfaces with high level programming language using Component Object Model (COM). Component Object Model (COM) is a binary interface standard for software components. It is used to enable inter-process communication.
- We have used python pywin32 package to interface with Thermal Desktop.

NASTRAN, SigFit and Code V

- NX NASTRAN is a Finite Element solver for stress, vibration, buckling, structural failure, heat transfer, acoustic and aeroelasticity analyses.
- Command line interface available to run in batch mode. Support for utilizing multiple processors/cores.
- SigFit allows optomechanical engineers to integrate mechanical analysis with optical analysis.
- It has a command line interface to run in batch mode
- Code V optical design software is used to model, analyze, and provide fabrication support for the development of optical systems.
- Code V can be run in both graphical command line interface

Pipeline Execution

- Pipeline is executed from command line as a python script

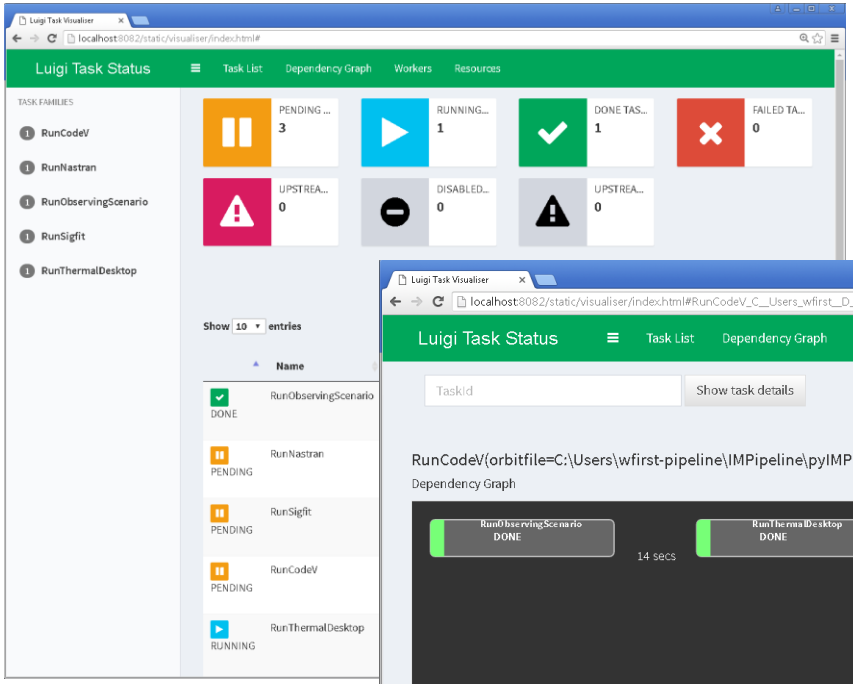
  > python impipeline.py RunCodeV --obsdate YYYYMMDD --ref-ra XXX.XXX --ref-dec XX.XXX --target-ra XXX.XXX --target-dec XX.XXX --initial-ra XXX.XXX --initial-dec XX.XXX --outpath XXXXXXXXX

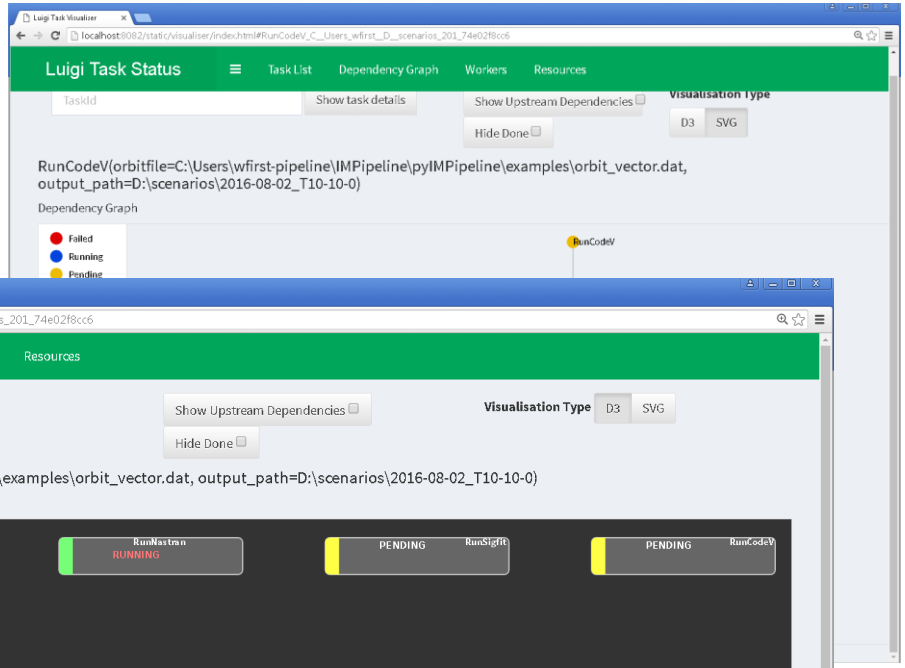
- Command line input parameters:
  - Mandatory
    - --obsdate : observing date in YYYYMMDD format
    - --ref-ra : RA of the reference star
    - --ref-dec : DEC of the reference star
    - --target-ra : RA of the target
    - --target-dec : DEC of the target
  - Optional
    - --initial-ra : RA of initial pointing
    - --initial-dec : DEC of initial pointing
    - --outpath : Output directory where outputs are written to

Pipeline Visualization

- Pipeline execution status can be checked using Luigi's web interface. Luigi runs a web server (Tornado) in the backend and displays task status in the web interface.

- Web interface can be accessed by going to following URL in web browser:
  http://localhost:8082/static/visualiser/index.html





Pipeline Output & Diagnostics

- If user does not define an output directory, current timestamp is used as directory name
- Although we are interested in wavefront errors, output files from Thermal Desktop, and NASTRAN are also saved for diagnostic purposes
- A consolidated log file (impipeline.log) is generated for each run. It includes messages from all the tasks, along with timestamps. It can be used to track down errors
- Creation of dummy file for each task signifies completion of that task to Luigi. So, the next time if you try to run the pipeline in the same directory where the pipeline ran successfully previously, Luigi will skip the execution
- Using Luigi check pointing, it is easy to re-run the pipeline from any intermediate task

Customizing Pipeline

- IMPipeline can be easily customized using configuration files
- Three configuration files –
  - impipeline.cfg : configure various tasks in the pipeline
  - luigi.cfg : configure Luigi framework itself
  - logger.cfg : configure logging capability of the pipeline
- Email notification. Sends out an automated email notification if the pipeline fails with an error.
- User level customization. Pipeline is developed to be run by multiple people involved in the project. The configuration files allows users to customize the pipeline run based on your individual requirements.

Verification and Validation of STOP Model

The recent study "Thermal system verification and model validation for NASA's cyrogenic passively cooled James Webb Space Telescope," by P. Cleveland and K. Parrish, 05ICES-236, SAE Aerospace, 2005 suggest the following three aspects be key ingredients in a thermal verification and model validation plan:

1. Margin requirements and tracking
2. Independent thermal modeling
3. Comprehensive thermal test program

To date, the following are how the above have and continued to be addressed in the V&V effort of the IMPipeline:

1. Flow down of top level requirements to the Coronagraph (CGI) modeling team from the WFIRST Observatory / TELESCOPE is communicated weekly via Integrated Modeling meetings to coordinate the workflow and design amongst JPL and GSFC engineering teams
2. JPL has developed its own internal model of the WFIRST Observatory / TELESCOPE /Collimator/ Coronagraph (CGI) independent of that created by GFSC
3. TVAC (thermal vacuum of the WFIRST TELESECOPE and sub-components such as CGI and Collimator are future work

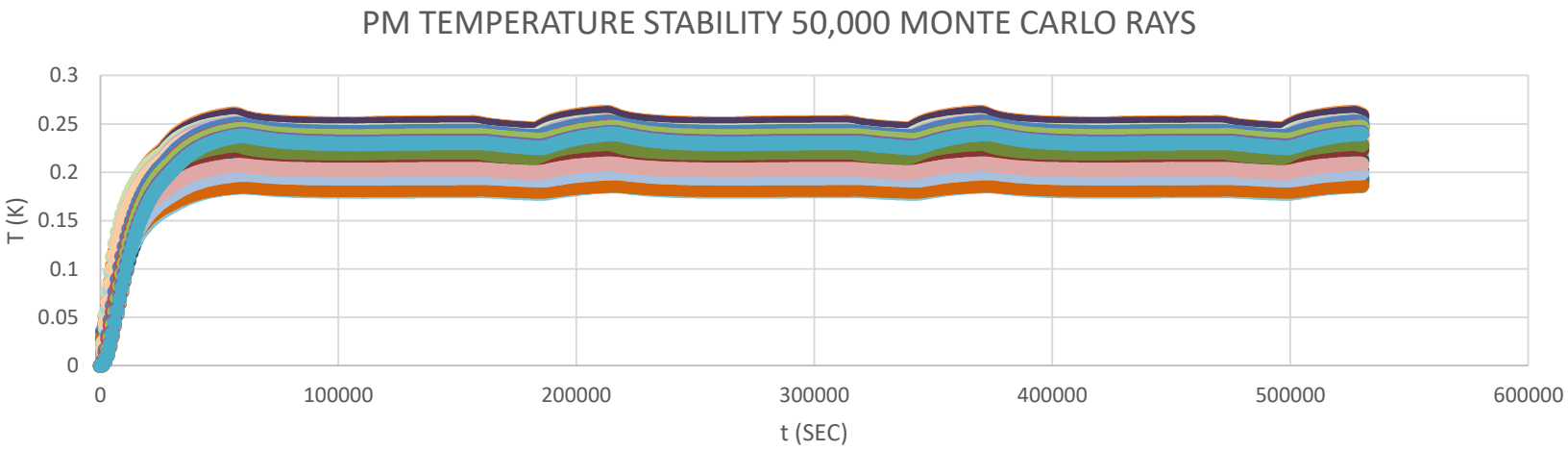
Verification and Validation of Models

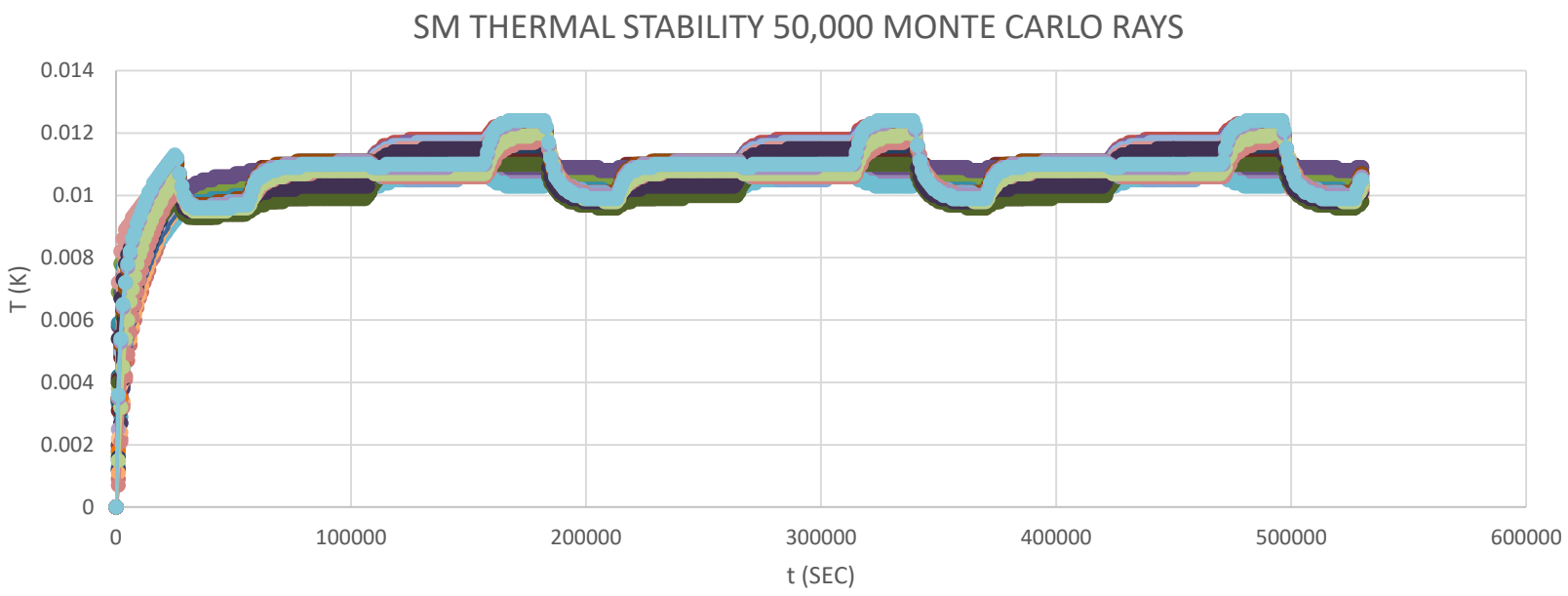
- Per "Verification and validation of simulation models," by J. Kleijnen, European Journal of Operational Research, Vol. 82, pp. 145-162, 1995

- The verification and validation of simulation models such as those used within the IMPipeline discussed herein involves the following activities:
  - Verification
    - **"Includes good programming practice (such as modular programming)"** such as that detailed in the Python script for the IMPipeline
    - **"Checking intermediate simulation outputs**" this has been and continues to be the standard daily work flow practice when dealing with the results of the IMPipeline
  - Validation
    - **"Comparing simulated and real data through simple tests such as graphical tests"** this is documented in the daily workflow of the IMPipeline activities by formulating and construction various in-situ, turn-key "Operational Scenario" which involve a series of yaw, pitch, roll and pointing maneuvers of the TELESCOPE / OBSERVATORY simulation model framework , data from each OS is viewed and post-processes via graphical results
    - **"Obtaining real world data"** (this will be future work)
    - **"Perform sensitivity analysis based on Monte Carlo sampling"** an example of this is shown on the next chart, whereby the number of Monte Carlo rays is varied to ascertain the impact on the RADK (Radiation Conductors) and model simulation run time

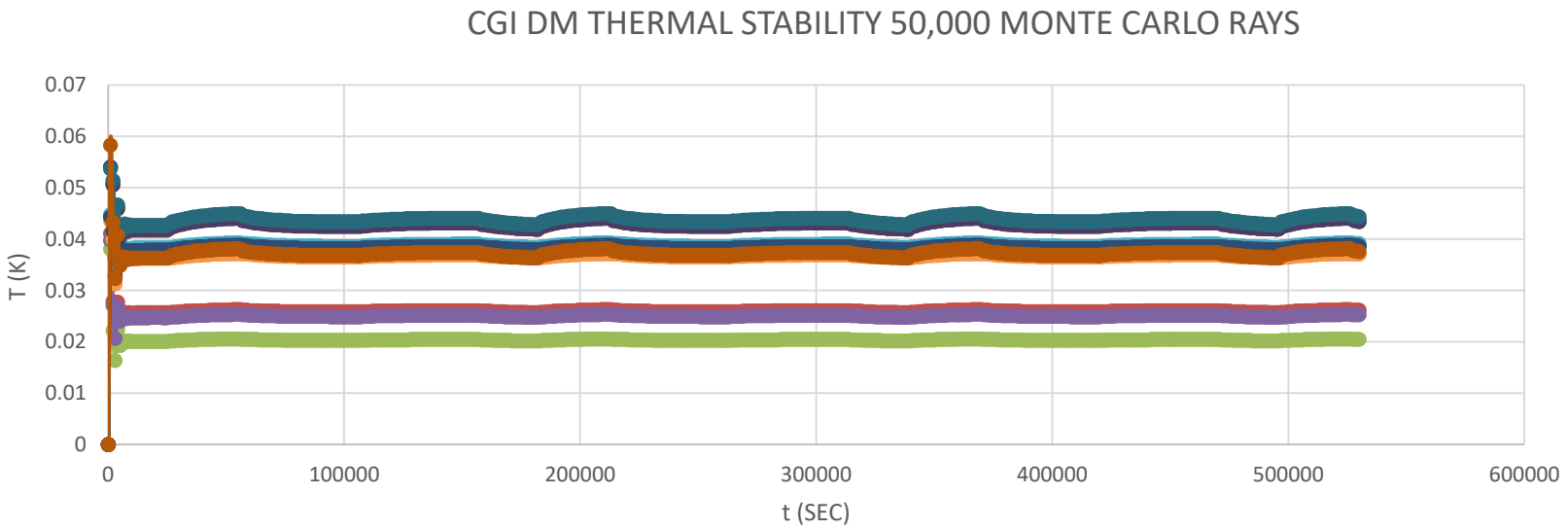
Preliminary Results - I

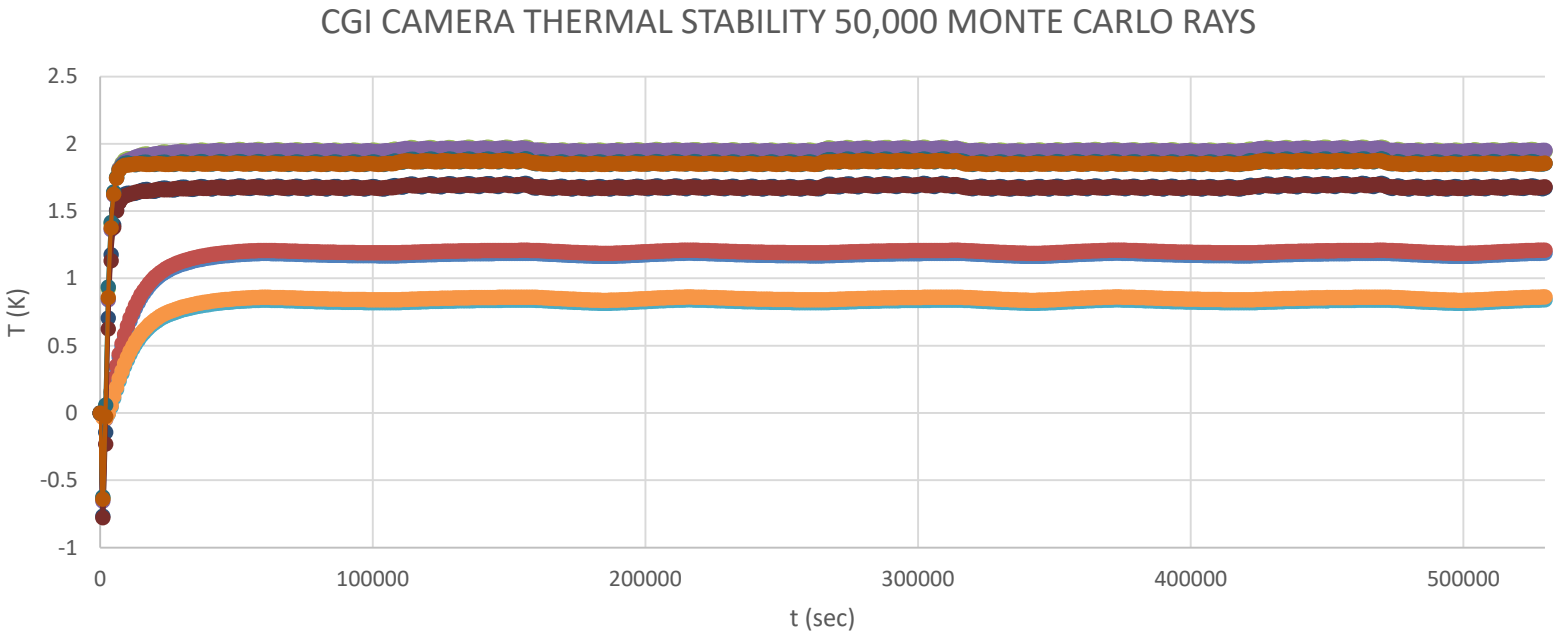
Exercise pipeline to investigate thermal stability of the instrument





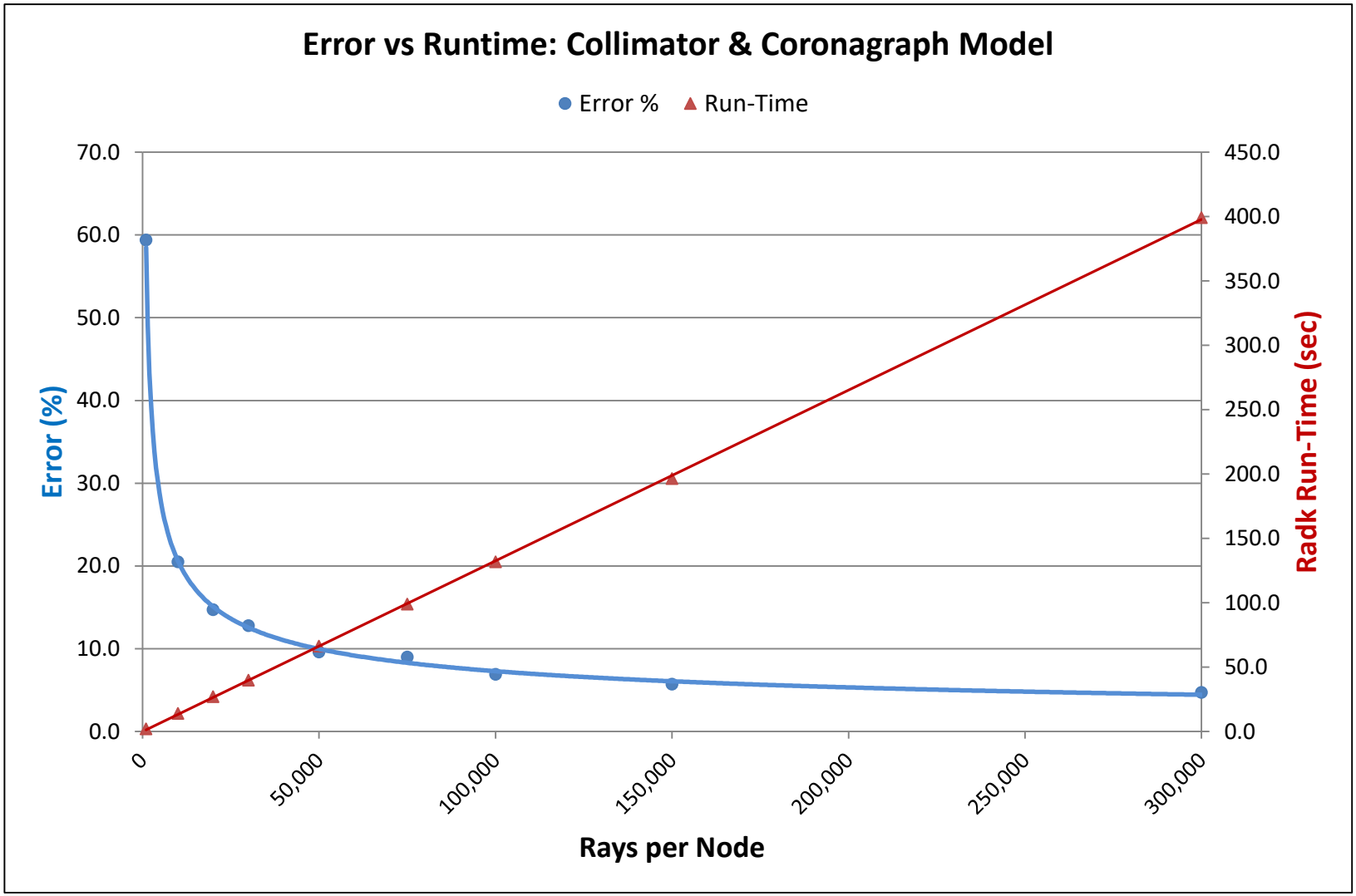
Preliminary Results - II





Monte Carlo Ray Tracing Trade Study

A trade study was performed to determine the optimal # of rays per node versus error and the associated runtime



Future Work

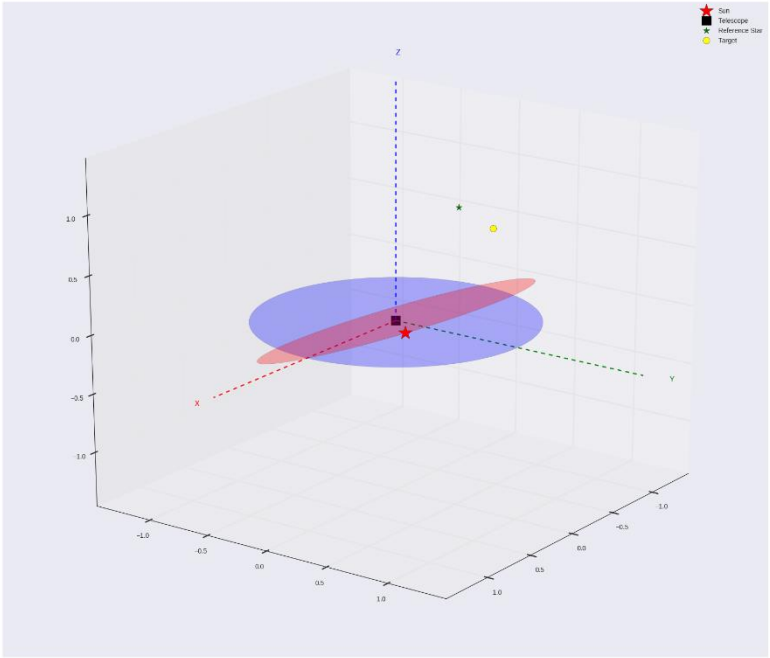
- Create a library of wavefront errors for various “Operational Scenarios”
- Validation of CGI performance using updated observatory STOP model
- Extend the IMPipeline beyond STOP analysis by Integrating the CGI wavefront propagation code in the pipeline to generate realistic speckle patterns.

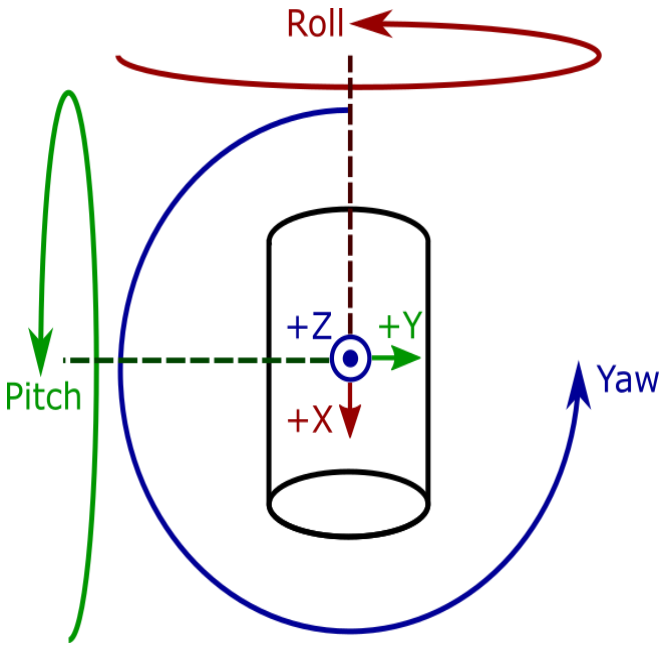
Thank You!

Backup Slides

Observing Scenario - I

Solar vector X, Y, and Z components comprise a unit vector pointing in spacecraft body-fixed coordinates to the center of the Sun.





Spacecraft coordinate frame and rotations (William Schulze)

Solar Vector X = sin(pitch)

Solar Vector Y = sin(roll)

Solar Vector Z = cos(pitch) * cos(roll)

Observing Scenario - II

- Determine position of Sun relative to Earth in ecliptic coordinate system
- Check if the target is within the Sun exclusion angle
- Transform position of reference star and target to ecliptic coordinate system
- Express all positions in heliocentric ecliptic coordinates
- Define spacecraft body centered coordinate system (BCCS)
- Determine transformation matrix between heliocentric coordinates and BCCS
- Initial orientation of spacecraft is +Z pointing towards the Sun and +X (boresight) direction pointing south out of the equatorial
- Slew to reference star
- Slew to target 2 and repeat steps as for target 1
- Reference Star Differential Imaging (RDI) roll
  - Roll about X for specified angle (+13 degrees to -13 degrees)
- Generate a text file with instantaneous time and Solar vector angles for telescope maneuver from initial state to target star